Homework 1

Due Date: Feb 15, 2007 (Thursday) in class.

Submission: Submit in class

Instruction: Turn in short, precise and concise answers. Turn in all Matlab/C /C++ code (a synopsis of it will do if it is large sized) that is requested or that you used to calculate the requested values. It should be clear what commands were used for getting answers for the programming questions. For this homework, MATLAB might be the easiest tool (preferred) but you are free to use whatever programming language you are comfortable with.

Grading: Maximum marks are 90. This will be scaled to account for 15% of your total course marks. No collaborations allowed. Homeworks are individual assignments and any attempts at plagiarism will be viewed very seriously.

Problem 1 (VC Dimensions)

Show that the VC dimension of a binary classifier f(**x**,**w**) which forms a rectangular shaped box aligned with the axes is four. ***Note:*** The sides of the rectangle can be moved parallel to the axes by changing the **w** parameter and the classifier outputs one when a point is inside the box and zero otherwise. ***[10 points]***
You may use a diagrammatic sketch to illustrate your solution.

Problem 2 (Least Squares and Unbiased Estimate)

Show that $\hat{\mathbf{w}}$ (*Least Squares Estimate*) is an Unbiased Estimate, i.e., $E(\hat{\mathbf{w}}) = \mathbf{w}$ ***[10 points]***

Problem 3 (Non-linear approximation with linear methods)

We will use the regression method to predict housing prices. The data is contained in the file 'housing.data' and the description of the data and the columns and is contained in the file 'housing.names'. They are provided on the course web page.

We will try to predict the median house value (the 14th, and last column of the data) based on the other columns.

(a) First we will use a linear regression model to predict the house values, using the squared error as criterion to minimize. In other words,

$$y = f(\mathbf{x};\hat{\mathbf{w}}) = \hat{w}_0 + \sum_{i=1}^{13} \hat{w}_i x_i \text{ where } \hat{\mathbf{w}} = \arg\min_w \sum_{t-1}^{n} (\mathbf{y}_t - f(\mathbf{x};\hat{\mathbf{w}})^2).$$

Write the following functions: ***[10 points]***

- A function that takes as input weights **w** and a set of input vectors $\{\mathbf{x}_t\}_{t=1,\ldots,n}$ and returns the predicted output values $\{y_t\}_{t=1,\ldots,n}$
- A function that takes as input training input vectors and output values and returns the optimal weight vector $\mathbf{w}_{\text{opt}}$
- A function that takes as input training set of input vectors and output values, and a test set input vectors, and output values, and returns the mean training error (i.e. average squared-error over all training samples) and mean test error.

(b) To test our linear regression model, we will use part of the data set as a training set and the rest as a test set. For each training set size, use the first lines of the data file as a training set and the remaining as a test set. Write a function that takes as input the complete data set and desired training data set size and returns the mean training and test errors ***[10 points]***

Submit the mean squared training and testing errors for each of the following training set sizes: 10, 50, 100, 300, 400.

(Validation: If for a sample size of 100, you get a mean training error of 4.15 and mean test error of 1328, you are doing well!!)

(c) What conditions must hold for the training input vectors so that the training error will be zero for any set of output values? ***[10 points]***

We will now use polynomial regression. We will predict the house values using the functions of the form: $f(\mathbf{x};\hat{\mathbf{w}}) = \hat{w}_0 + \sum_{i=1}^{13}\sum_{d=1}^{m} \hat{w}_{i,d} x_i^d$. Here, again the weights are chosen so as to minimize the mean squared error of the training set.

(d) Modify your programs from the previous section such that it takes as input also a maximal degree m and returns the training and testing error under such a polynomial regression model. ***[10 points]***

NOTE: When the degree is high, some features will have extremely high values while the others will have very low values. This causes severe problems with matrix inversions and yields wrong answers. To solve this, you will have to appropriately scale each feature included in the regression model to roughly bring all the features to roughly the same magnitude. Be sure to use the same scaling for the training and testing examples. (MATLAB matrix and vector operations can be useful for doing such scaling).

(e) Prove that the scaling of features does not change the regression predictions. ***[10 points]***

That is, given training feature vectors and output values $\{\mathbf{x}_t, y_t\}_{t=1,\ldots,n}$ and a test input vector $\mathbf{x}_{test}$ and the scaling factors $\{\alpha_i\}_{i=1,\ldots,13}$, we would like to prove that the predictions of the test output value would be the same if we trained a linear regression on $\{\tilde{\mathbf{x}}_t, y_t\}_{t=1,\ldots,n}$ where $\tilde{\mathbf{x}}_{t,i} = \alpha_i \mathbf{x}_{t,i}$ and predicted on $\tilde{\mathbf{x}}_{test} = \alpha_i \mathbf{x}_{test,i}$.
[We need to prove this only for linear models (maximum degree one) and that is what we require you to prove. The actual result we use is for 'extended' features x_i^d , which is what we actually do since this is just linear regression using these 'extended features'.]

(f) For training set size of 400, provide the mean squared training and test errors for maximal degrees of zero through ten. ***[10 points]***
(Validation: For maximal degree 2, you should get training error of 14.5 and test error of 32.8)

(g) Explain the qualitative behavior of the test error as a function of polynomial degree. Which degree seems to be the best choice? ***[5 points]***

(h) Prove (in two sentences) that the training error is monotonically decreasing with the maximal degree m. This is, the training error using the higher degree and the same training set is necessarily less than or equal to the training error using a lower degree. ***[5 points]***